



유튜브 기반 스터디 플랫폼



INDEX

01

배경 및 소개

1. 팀 소개
2. 개발 배경
3. 프로젝트 소개
4. 역할 배분

02

기술

1. 사용 기술
2. ERD
3. 와이어 프레임
4. 주요 기술

03

개발 현황

1. 개인 진행
2. 팀 진행

04

개발 문화 및 계획

1. 브랜치 전략
2. 컨벤션 전략

05

Q&A

1. 예상 질문
2. Q&A

배경 및 소개

팀 & 프로젝트

01



박서영



이태희



정준호

1) 팀 소개



Team PEACE

6명의 조각이 모여 하나의 작품을 완성하겠다!
평화로운 팀플의 의미도 담고 있는 중의적인 의미

팀장 | 박서영 팀원 | 김동준 배서연 이태희 정준호 정채린

2) Why?

실시간 정보 공유의 부재 + 혼자 공부하는 외로움 = 인터넷 강의 완강률 5-6%



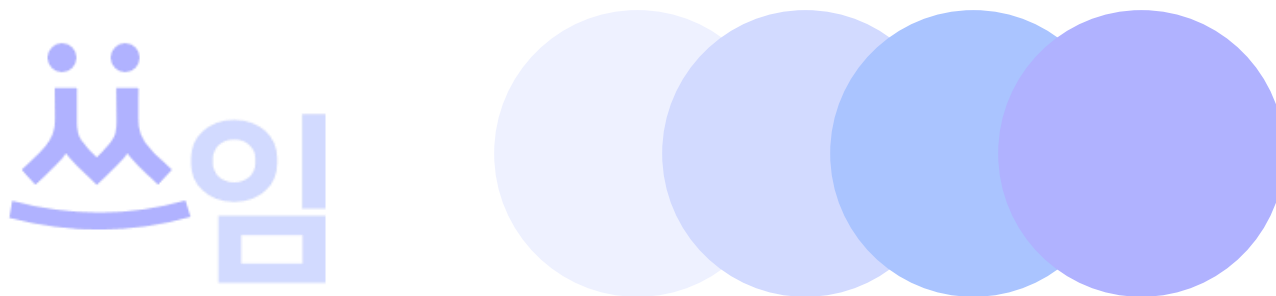
1) Why?

인터넷 강의로 오프라인 강의와 비슷한 사용자 경험을 주고 싶어!



3) 프로젝트 소개

“함께 온라인 강의를 들으며 실시간으로 타인과 교류하는 스터디 플랫폼”



[프로젝트 명] 스터디 모임의 줄임말

[핵심 컨셉] 스터디 모임을 만들어 강의를 함께 보고 실시간으로 질문을 주고 받을 수 있는 플랫폼

[프로젝트 심볼] 아래쪽을 보았을 때는 웃는 표정, 위쪽을 보았을 때는 사람이 모인 형태

[테마 컬러] Melrose #B1B2FF

4) 역할 배분



김동준

[Frontend]

UI/UX 담당

기획 및 디자인

와이어프레임 및 API설계



박서영

[Backend]

기술고문

API 설계 및 관리

모델링 및 DB 설계



정준호

[Backend]

실시간 데이터 및 DB 담당

모델링 및 내부 DB 설계

API 설계 및 관리



정채린

[Frontend]

기획 및 디자인

UI/UX 담당

와이어프레임 및 API설계



배서연

[Backend]

와이어프레임 및 API 설계

기획 및 디자인

아키텍처 설계



이태희

[Backend]

DevOps, 배포

CI/CD 파이프라인 구축

컨테이너 오케스트레이션

기술

사용 & 주요 기술

02



배서연

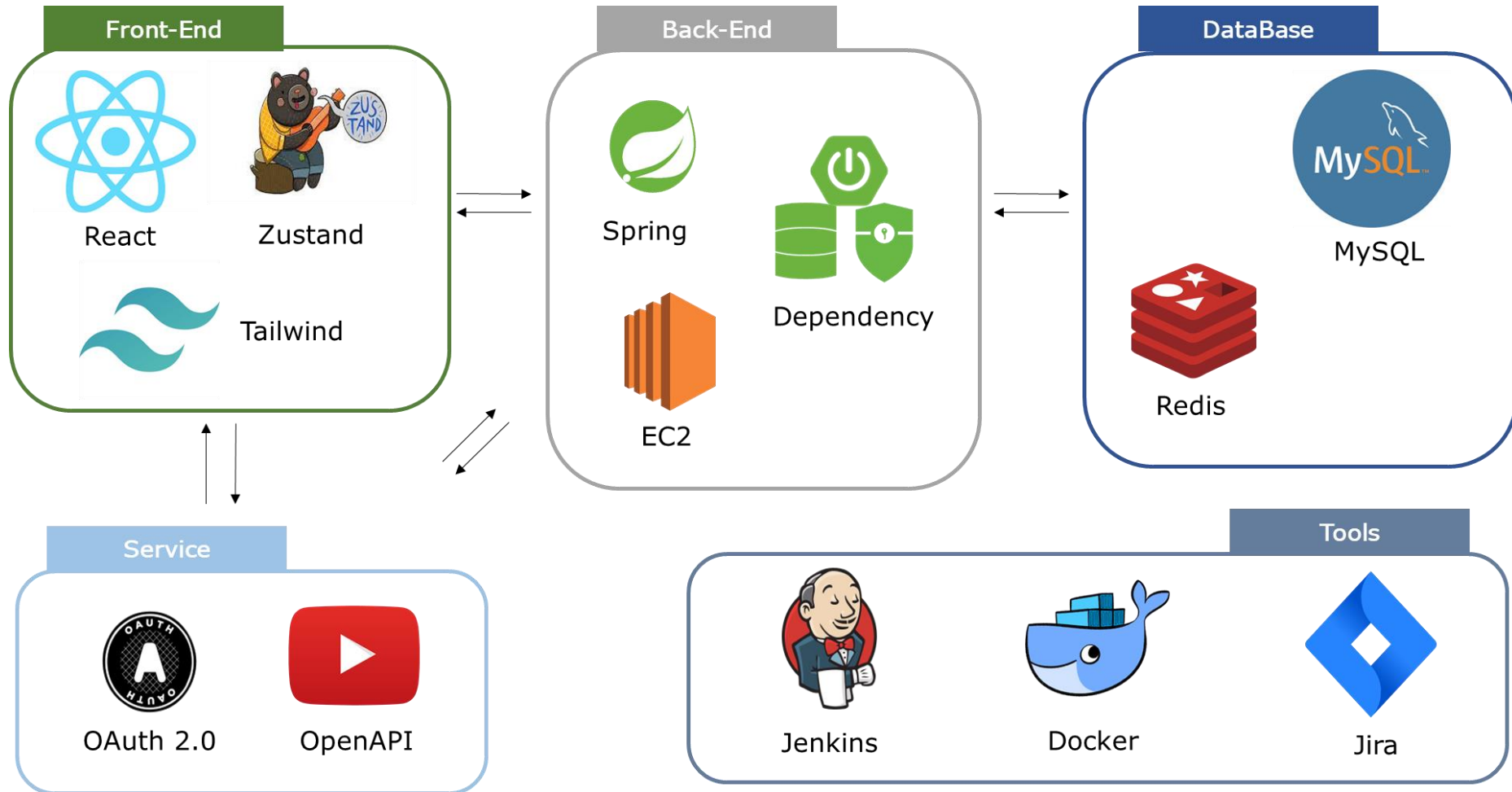


김동준

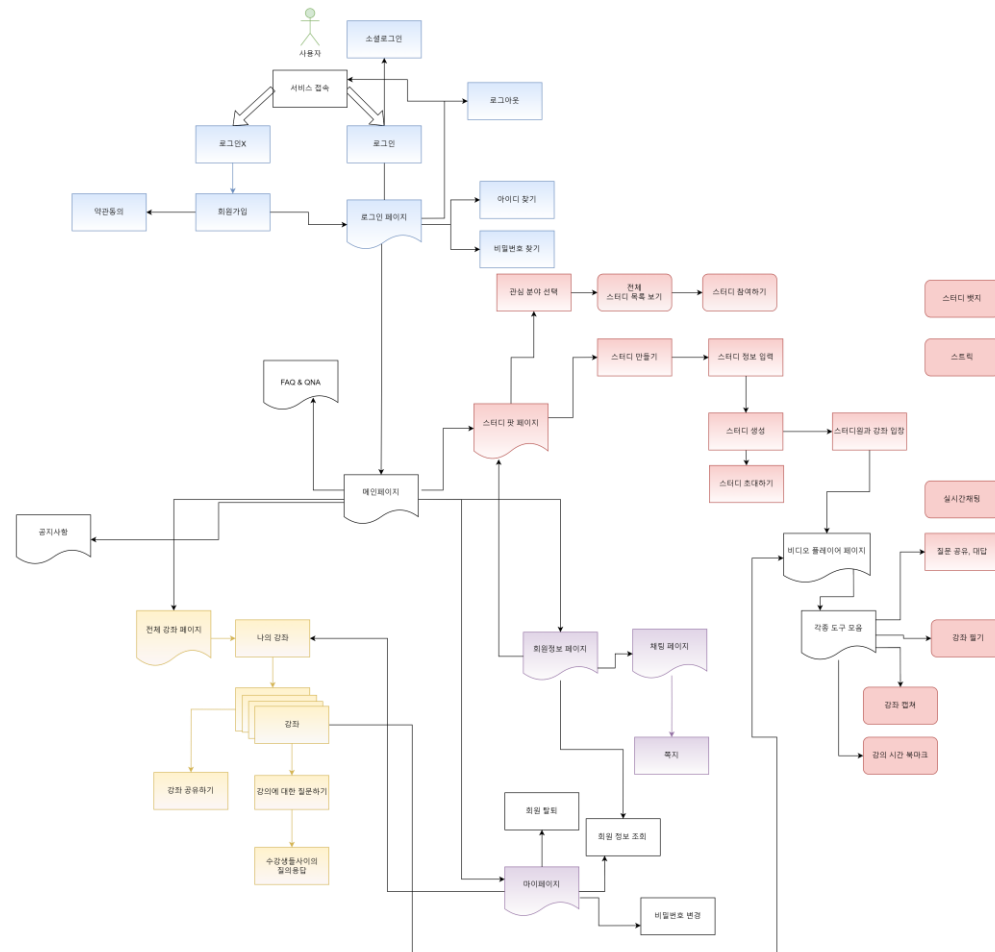


정채린

1) 기술 스택



2) 아키텍처 (drawio)



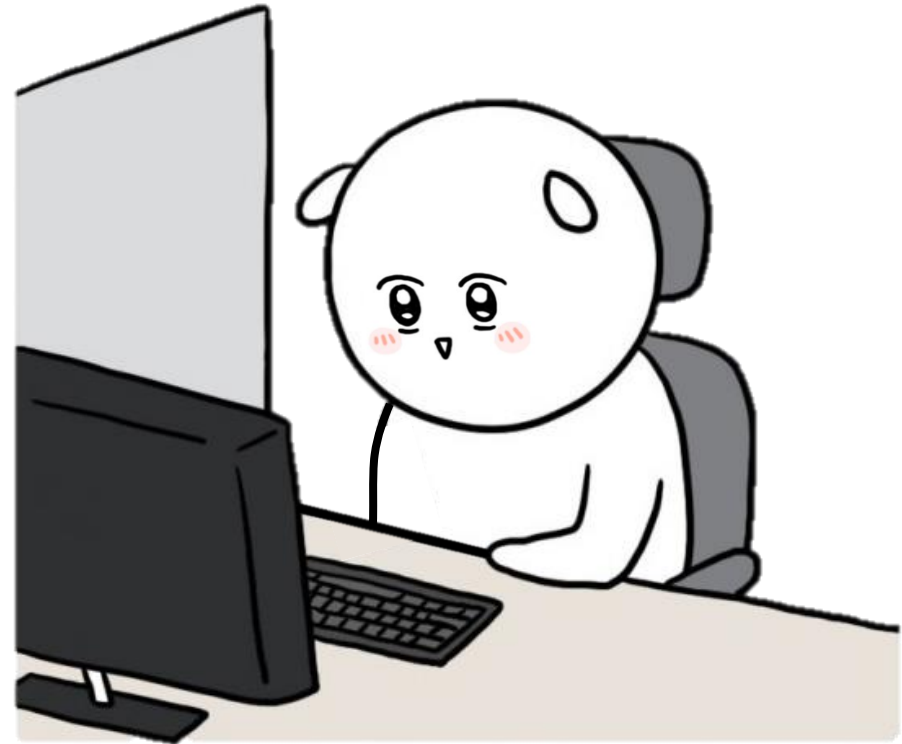
사용자 요구사항 정의서 기능 명세서

기능 명세서

[illegible]

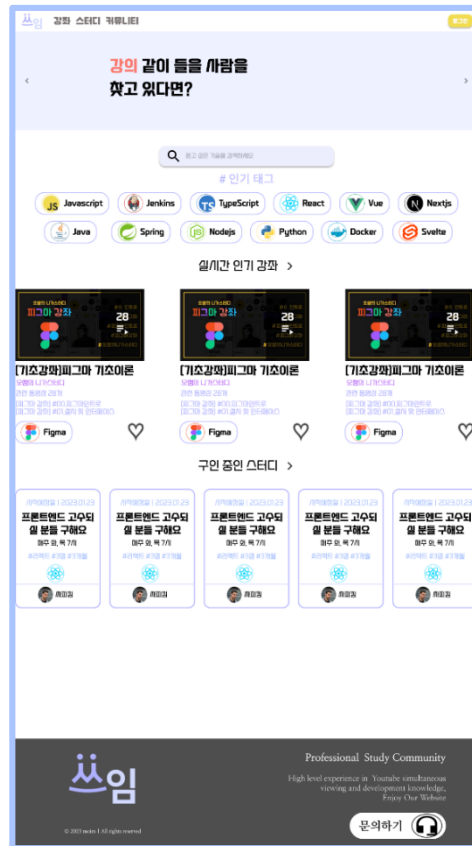
3) 와이어 프레임

스터디 진행 & 강의 시청 시
제공되는 사이트 내 서비스 기능
활용도 향상

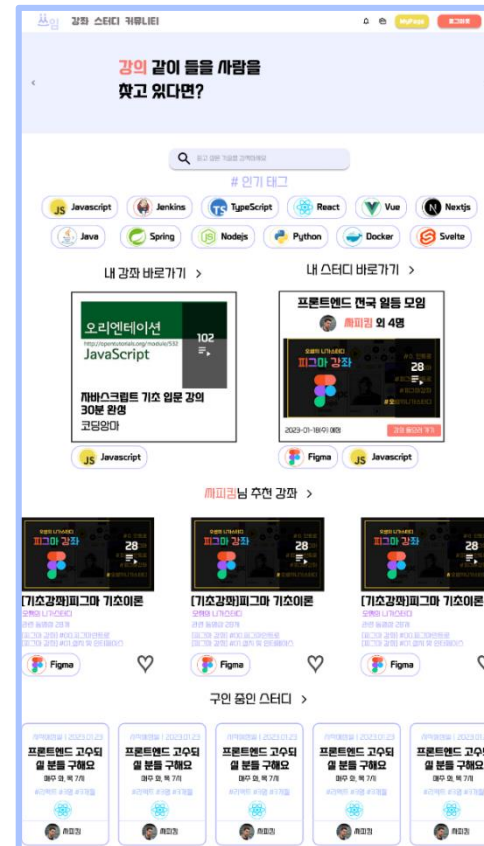


3) 와이어 프레임 - 메인 페이지

로그인 전

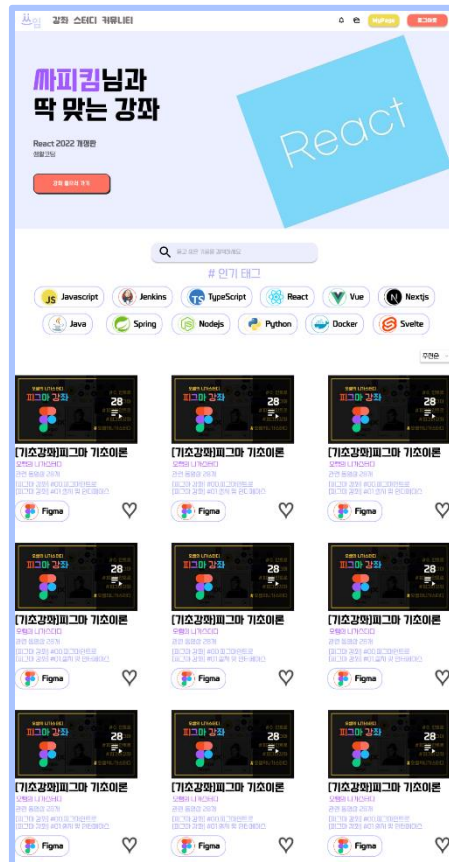


로그인 후

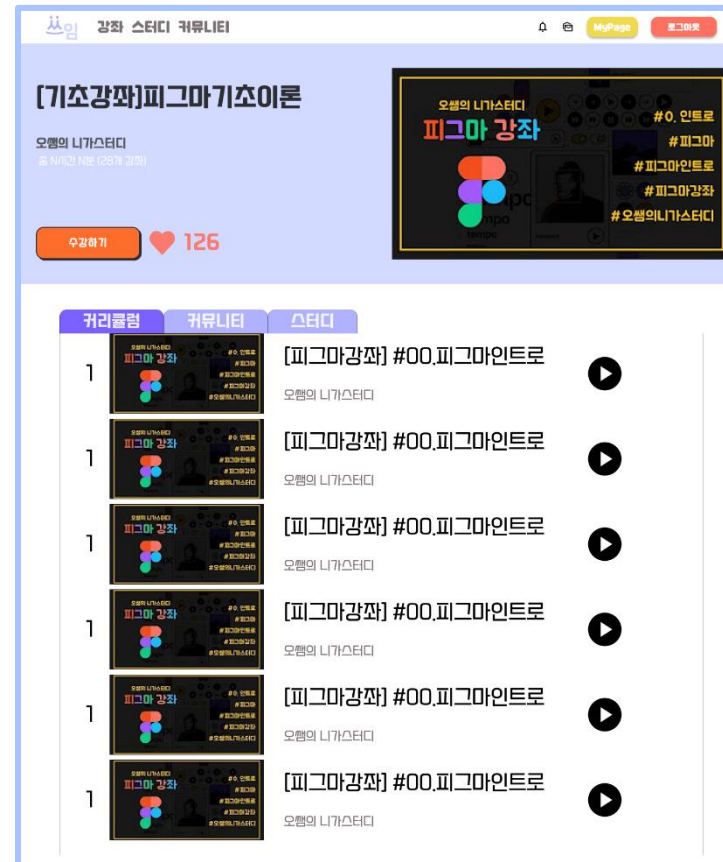


3) 와이어 프레임 - 강좌 페이지

강좌 페이지



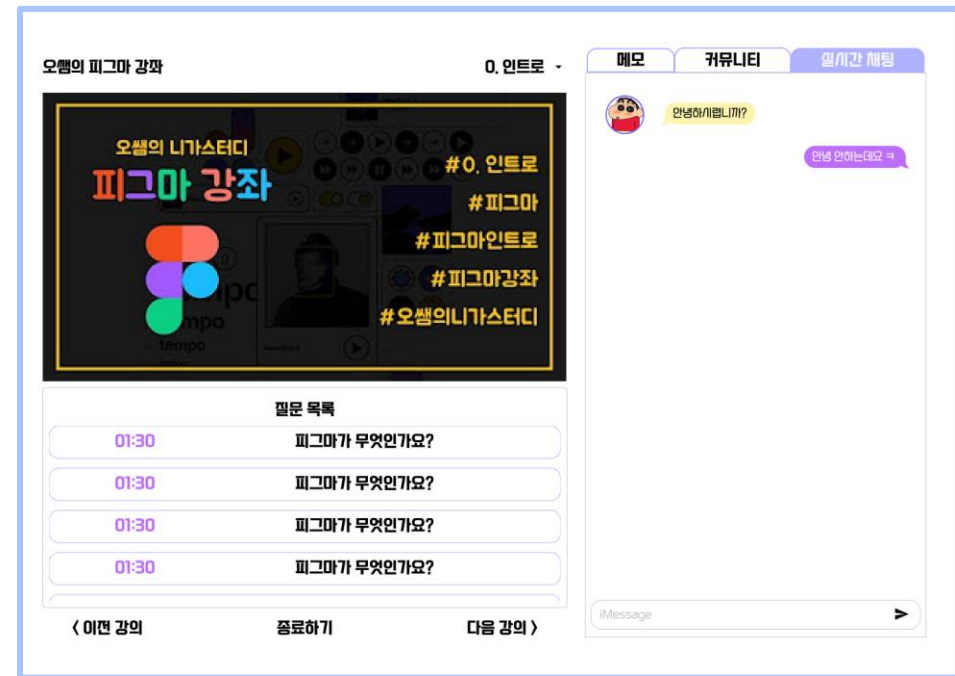
강좌 상세 페이지



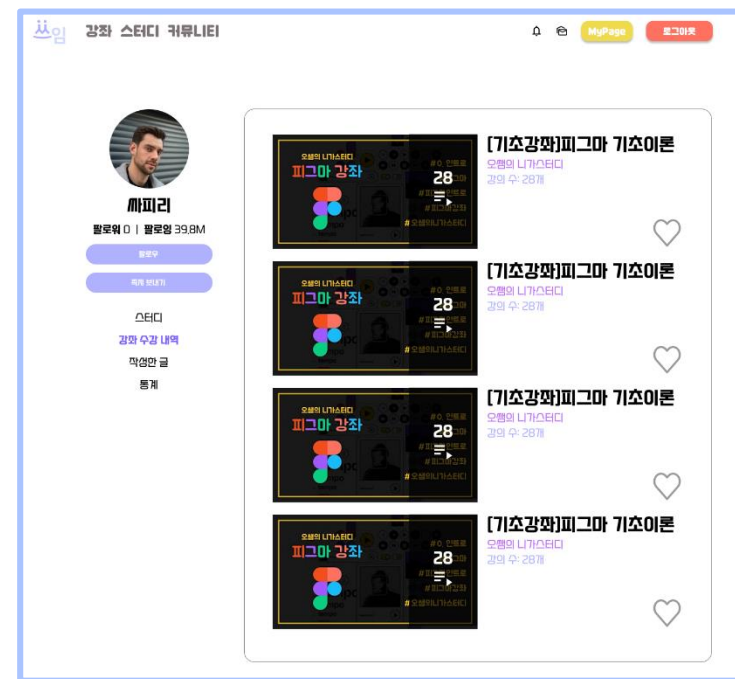
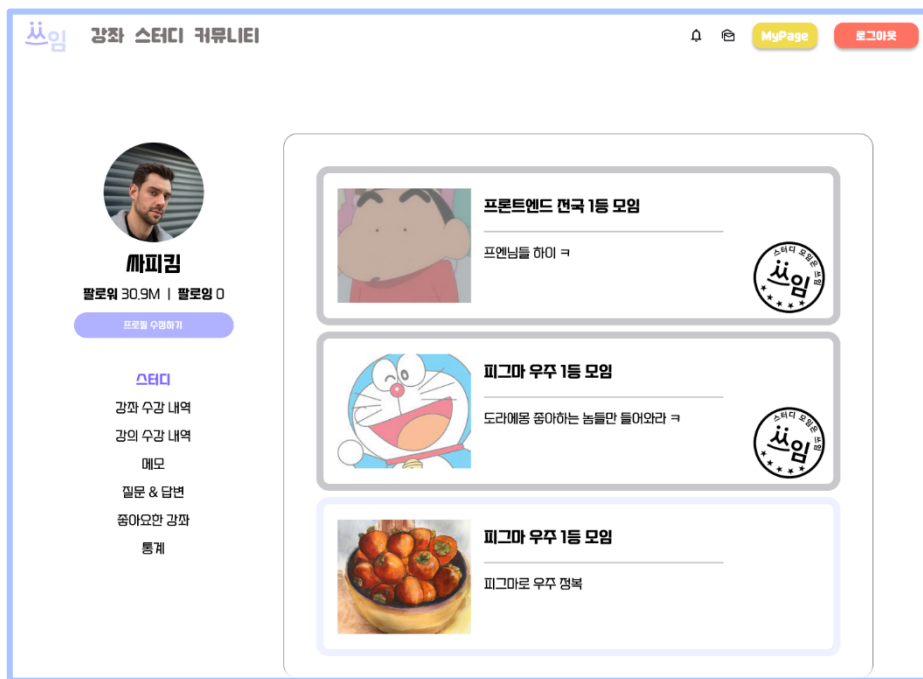
멤버용



3) 와이어 프레임 - 플레이어

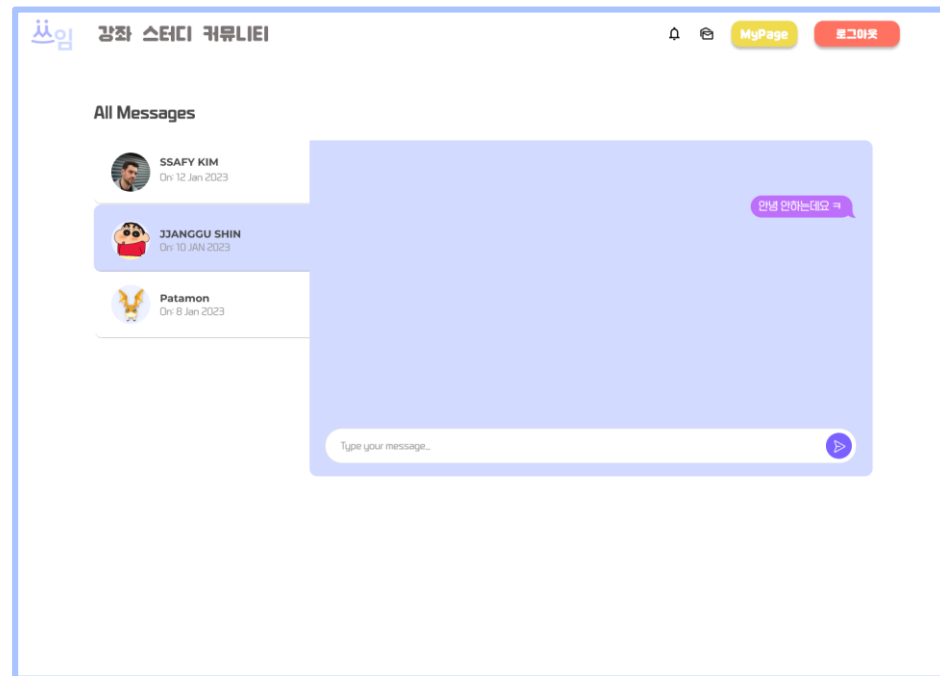


3) 와이어 프레임 - 마이페이지



3) 와이어 프레임 - etc

쪽지

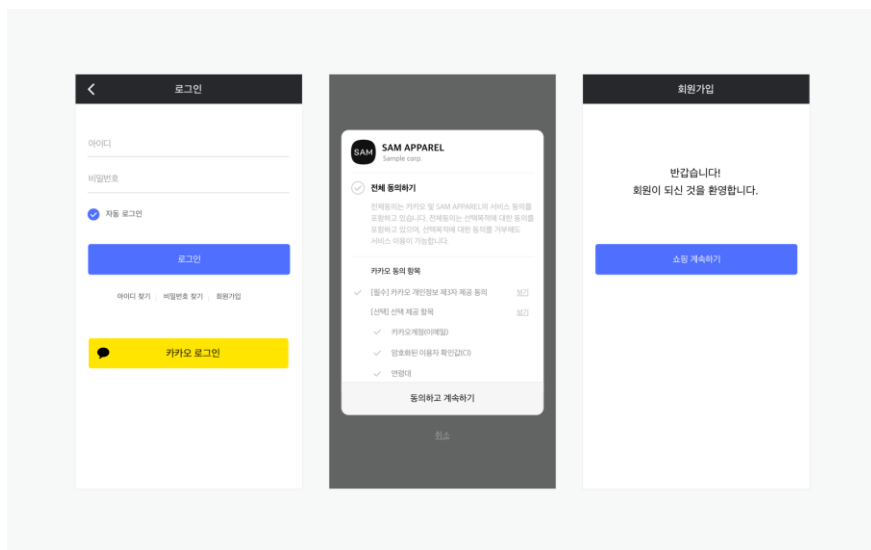


알림



4) 주요 기술 - 회원 관리

OAuth Token을 기반으로 JWT Token을 발급하여 사용자 권한 관리
WAS 버전 업데이트로 인해 배포가 중단되지 않도록 로드 밸런싱 계획

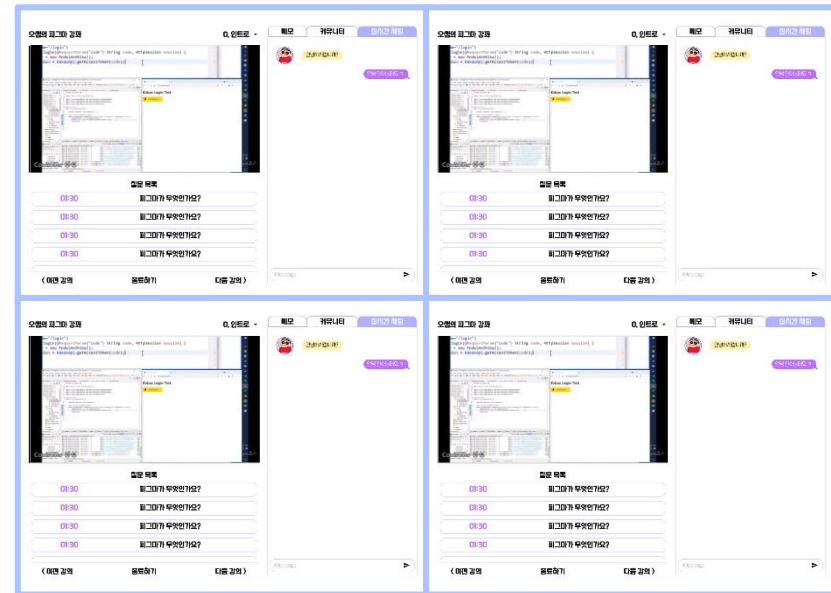


OAuth 회원



4) 주요 기술 - 동시 시청

Youtube API와 Redis를 함께 사용해 사용자 시청 시간 동기화



개발 현황

진행 상황

03



배서연



김동준



정채린

1) 개인 진행 상황



2) 팀 진행 상황

SERVER

젠킨스로 도커 기반 CI/CD 파이프라인 구축 완료

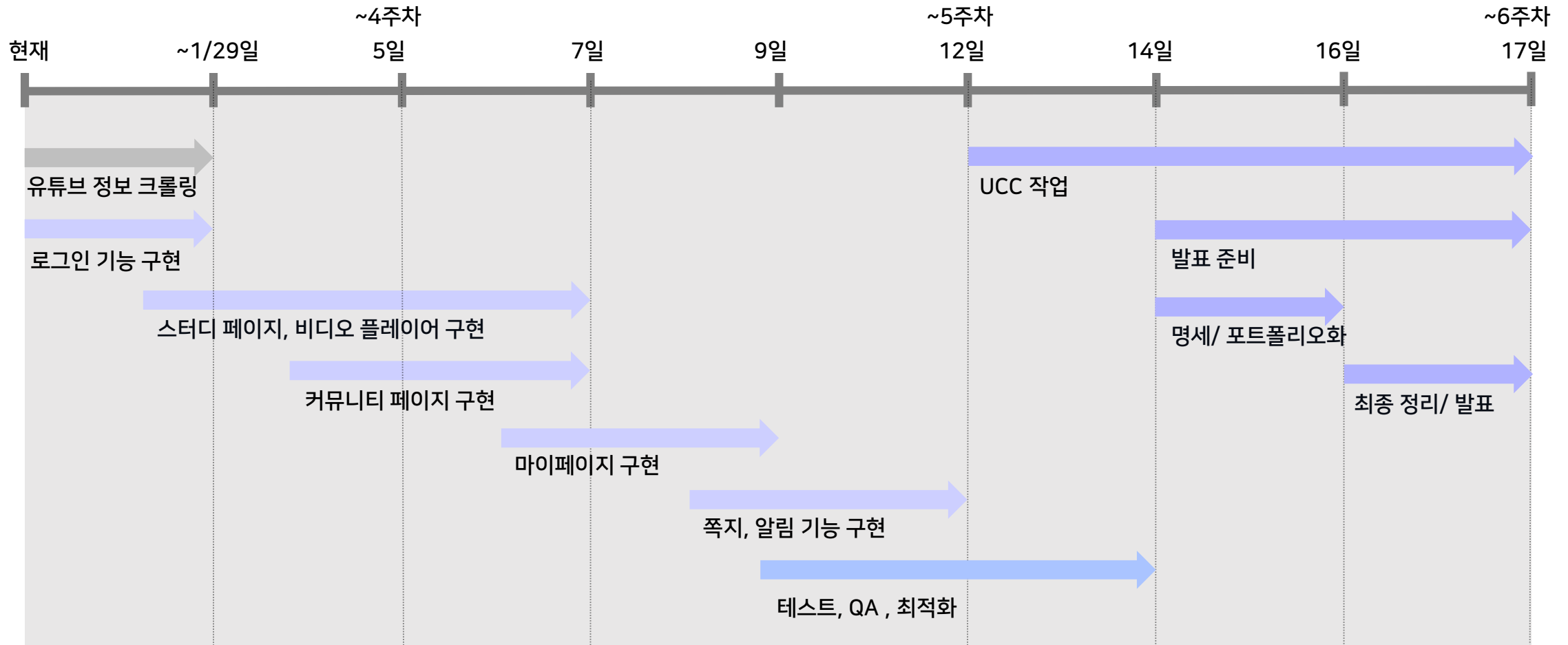
Front

커뮤니티 페이지
전체적인 디자인 완료
스터디 생성 폼 및
구인 목록 출력

Back

프로젝트 구조 생성
유튜브 API 및
카카오 로그인 개발 중

3) 향후 계획



개발 문화 전략과 규칙

04

배서연

김동준

정채린

1) 일정 관리 - 지라

■ [FE] 커뮤니티 페이지 더미데이터 제작

커뮤니티



■ [BE] JPA repo, DTO 의논, 역할분담

설계



■ [FE] 커뮤니티 페이지 메인화면 구성

커뮤니티



■ [FE] 커뮤니티 페이지 기타 화면 구성

■ [FE] 메인 - 코스페이지 구성 및 더미데이터 작성

■ [FE] 메인 - 코스페이지 구성

■ [BE] JPA repo, DTO, ERD 의논, 역할분담

■ [FE] 스터디 등록 폼 제작 및 제출한 정보 스터디 구인 페이지에 띄우기

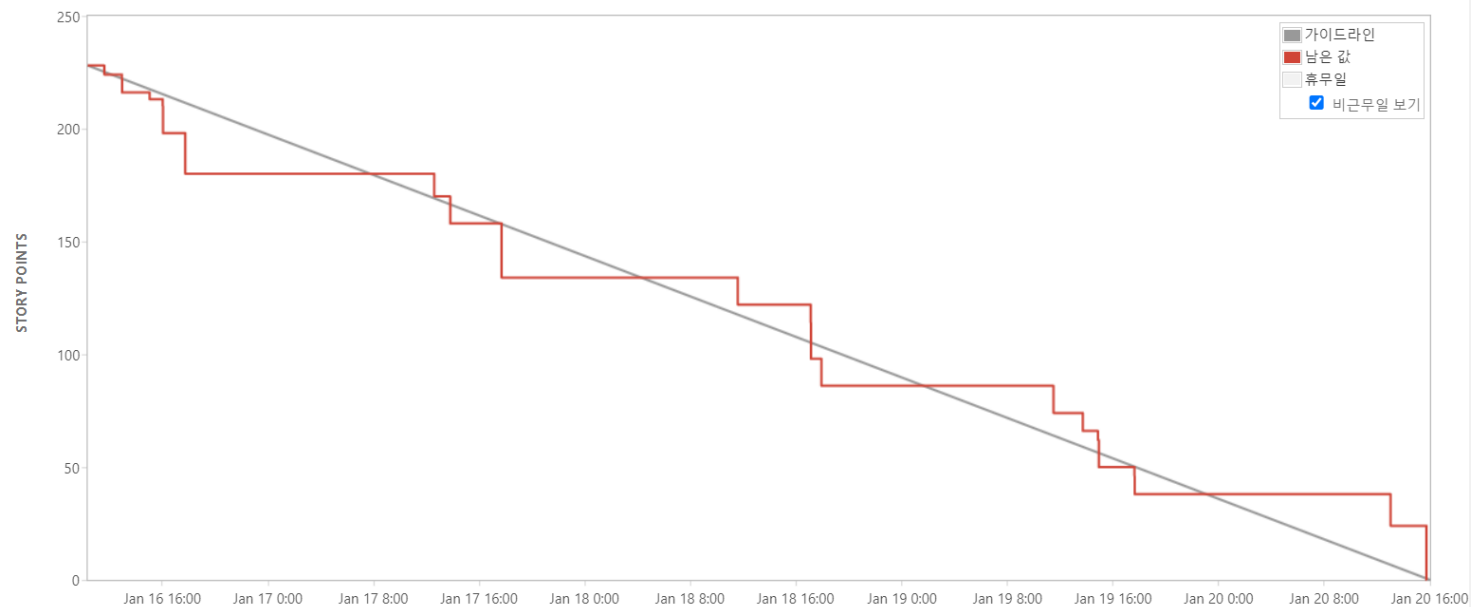
■ [FE] 스터디 구인 디테일 페이지 띄우기

■ [FE] 구글 로그인 프론트엔드 부분 구현

■ [FE] 메인 - 코스페이지 마무리

■ [공통] 데일리 스크럼 및 마일스톤 제작

■ [FE] 커뮤니티 페이지 다시 확인



1) 일정 관리 - 회의록

1월 9일

- 프로젝트 기획 고민에 대한 상담 진행
- 깃, 지라 학습
- 프로젝트 주제 선정: 온라인 학습 스테디
- EBS API 활용 문의

1월 10일

- 지라 사용법 익힘
- MM, GitLab 연결
- EBS 측에 전화 문의 - 부정적인 답변 받음
- Use Case Diagram 브레인스토밍
- 깃 커밋 컨벤션 지정

1월 11일

- EBS 컨택
- 프로젝트 목적 구체화
- 피그마 작업 시작

1월 12일

- 요구사항 정의서 작성
- 기능 명세 및 우선 순위

1월 13일

- 상호 평가
- ERD 초안 작성
- 와이어프레임 초안 작성

2023. 01. 16

- JIRA 주간 스프린트 작성
- 팀 미팅 후 ERD 수정
- 피그마 와이어프레임 구체화
- 스테디 개설 및 스테디 페이지 구
- 개인 학습

2023. 01. 17

- 피그마 와이어프레임 구체화
- ERD 수정
- 로고 만들기
- 개인 학습

2023. 01. 18

- 피그마 와이어프레임 구체화
- ERD 구체화
- FrontEnd 사용 툴 선정
- 시퀀스 다이어그램 간단 회의
- 개인 학습

2023. 01. 19

- 피그마 마무리
- ERD 마무리
- 웹소켓 채팅 구현 시도
- FE 초기 세팅 및 네비게이션바 생성
- 기능명세서 정리
- 발표 ppt 제작 시작
- 개인 학습

2023. 01. 20

- FrontEnd
 - 피그마 to React 해보기
 - Component 구조화
- BackEnd
 - Git Branch 전략 수립
 - 프로젝트 구조 정리

2023. 01. 25

- Team
 - PPT제작
 - 지라 3주차 스프린트 완성
 - 마일스톤 작성
- FrontEnd
 - 커뮤니티 페이지 구성
 - 스테디 페이지 구성
- BackEnd
 - Entity, Repository, ERD, DTO 의논
 - 자동 배포 설정 완료
 - YouTube API 공부



📄 Add cover 🗨 Add comment

회의록

+ :: 📄 Table +

Filter Sort 🔍 📄 ... New ▾

회의록 ...

Aa 회의 제목	🕒 회의 유형	📅 회의 날짜	☰ 회의 주제	+ ...
📄 1차 아이디어 회의	일일 회의	January 9, 2023	프로젝트 주제 선정	
📄 2차 아이디어 회의	일일 회의	January 11, 2023	스테디란 무엇인가	
📄 3차 아이디어 회의	일일 회의	January 12, 2023	기획 세분화하기	
📄 4차 아이디어 회의	일일 회의	January 13, 2023	프로젝트 이름, 컬러팔레트	
📄 1차 일정 회의	주간 회의	January 16, 2023	일주일 계획 정하기	
📄 5차 아이디어 회의	일일 회의	January 16, 2023	스테디 세부 사항	
📄 6차 아이디어 회의	일일 회의	January 17, 2023	로고 정하기, 스테디 내의 소통법	
📄 7차 아이디어 회의	일일 회의	January 18, 2023	시퀀스 다이어그램	
📄 8차 아이디어 회의	일일 회의	January 19, 2023	알람 유형	

2) 개발 문화 - 컨벤션

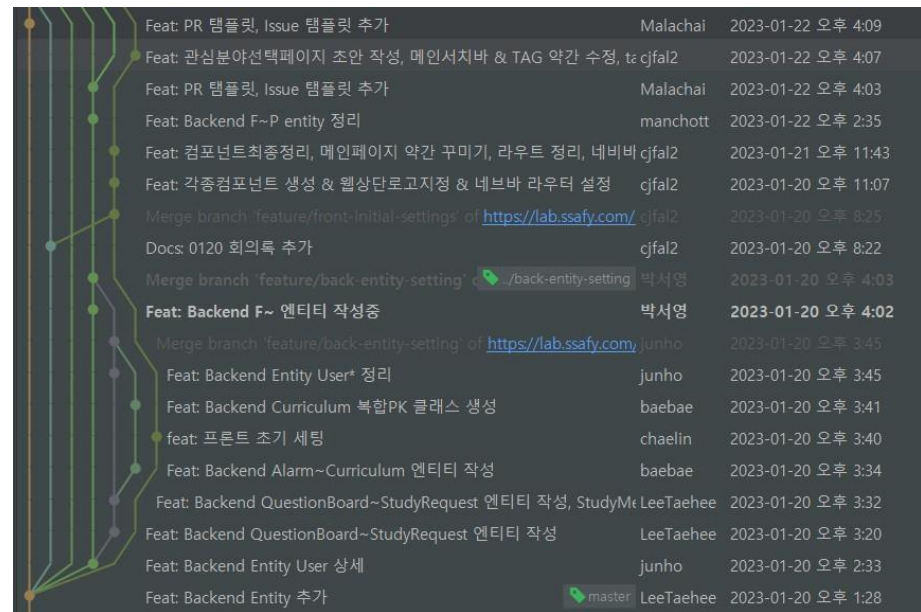
master

└ develop : feature들을 머지하는 브랜치

└ feature/[back or front]-[영문 기능]

: 구현해야하는 기능들을 개발하는 브랜치

: 기능 별로 관리



2) 개발 문화 – 컨벤션

깃

1. 커밋 유형 지정

- 커밋 유형은 영어 대문자로 작성하기

커밋 유형	의미
Feat	새로운 기능, 파일 추가
Fix	버그 수정
Docs	문서 수정
Style	코드 formatting, 세미콜론 누락, 코드 자체는 경우
Refactor	코드 리팩토링
Test	테스트 코드, 리팩토링 테스트 코드 추가
Chore	패키지 매니저 수정, 그 외 기타 수정 ex)
Design	CSS 등 사용자 UI 디자인 변경
Comment	필요한 주석 추가 및 변경
Rename	파일 또는 폴더 명을 수정하거나 옮기는 것
Remove	파일을 삭제하는 작업만 수행한 경우
[BREAKING CHANGE]	커다란 API 변경의 경우
[HOTFIX]	급하게 치명적인 버그를 고쳐야 하는 경우

2. 제목과 본문을 병행으로 분리

- 커밋 유형 이후 제목과 본문은 한글로 작성하여 내용이
- 본문에는 변경한 내용과 이유 설명 (어떻게보다는 무엇

3. 제목 첫 글자는 대문자로, 끝에는 . 금지

4. 제목은 영문 기준 50자 이내로 할 것

5. 자신의 코드가 직관적으로 바로 파악할 수 있도록

6. 여러가지 항목이 있다면 글머리 기호를 통해

명명 규칙

소문자(Lowercase)

데이터, 뷰, 컬럼을 비롯한 모든 식별자들은 소문자로 작성하는 게 좋습니다. 별자 이름들을 사용하는 건 좋지 않습니다. 이유는 예약어들과 구분짓기 위해서 데이터베이스를 생성하면 소문자로 출력되는 것도 이런 규칙을 지키는 게 0 ex) `first_name` 으로 사용하는 것이 `First_Name` 으로 사용하는 것 보다 좋

- 이후 데이터, 뷰, 컬럼들을 모아서 object라고 표현하도록 하겠습니다.

데이터 타입을 이름으로 정하는 것은 피하자

object에 type을 이름으로 사용하면 안됩니다. 특히 컬럼명도 명사로 지어(명을 사용하는 것은 좋지 않습니다.

ex) `text`, `timestamp` 등은 좋지 않은 컬럼명입니다.

복합어구에는 _를 사용하자

여러 글자가 합쳐져 만든 복합어구에는 `_` (snake_case)를 사용하는 게 좋습지 않습니다.

ex) `wordcount`, `wordCount` 보다는 `word_count`, `team_member_id` 가 좋습니다

축약어보다는 풀네임을 사용하자

Object 이름들은 약어를 사용하기 보다는 풀네임을 사용하는게 좋습니다. 스는 30자 이상의 Object 이름을 설정할 수 있도록 지원하고 있습니다.

ex) `middle_name` 이 `mid_nm` 보다 좋습니다.

약어를 사용해야 할 때는 공통적인 약어를 사용하자

어쩔 수 없이 약어를 사용해야한다면, 공통적으로 사용하는 약어를 사용하

- 저는 <https://www.curiousstore.com/#/util/naming> 이 사이트를 자주 0

단수 명사

table, view 등은 단수의 명사를 가지는게 좋습니다.그 이유중 첫번째는 일과의 관계를 맺을 때 1개의 row와 관계를 맺기때문이며, 두번째는 어플리케이션 혼란이 없기 때문입니다.

ex_1) teams보다는 team으로 테이블이름을 짓는게 좋습니다.ex_2) team_n `teamMember`가 되면 되고, python에서는 `team_member`로 사용하면 됩니

DB

코드 - 공통

“ 문자열을 처리할 때는 쌍따옴표를 사용하도록 합니다.

; 문장이 종료될 때는 세미콜론을 붙여줍니다.

함수명, 변수명은 카멜케이스로 작성합니다.

가독성을 위해 한 줄에 하나의 문장만 작성합니다.

주석은 설명하려는 구문에 맞춰 들여쓰기 합니다.

```
// Good
function someFunction() {
  ...

  // statement에 관한 주석
  statements
}
```

연산자 사이에는 공백을 추가하여 가독성을 높입니다.

```
a+b+c+d // bad
a + b + c + d // good
```

코마 다음에 값이 올 경우 공백을 추가하여 가독성을 높입니다.

```
var arr = [1,2,3,4]; //bad
var arr = [ 1, 2, 3, 4]; //good
```

코드 - 프론트 & 백

Naming 규칙

- 폴더이름은 소문자만! (snake case)
- tsx 파일은 pascal case (ex. `LogInMain.tsx`, `LogOutMain.tsx`)
- 함수명, 변수명은 camel case (ex. `onClickNumber()`)

Coding Convention

- 기본 틀

```
export default function tsx페이지명() {
  return (
    <>

    </>
  );
}
```

vercel 확장 · Prettier · Code formatter

google/google-
java-format



Reformats Java source code to comply with Google
Java Style.

80
Contributors

929
Used by

5k
Stars

787
Forks



Thank You.



Q & A

질문 시간

05



배서연



김동준



정채린